California State University, San Bernardino CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2002

Logserver monitor for managing log messages of applications

Lilin Zhu

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd-project

Part of the Computer Engineering Commons

Recommended Citation

Zhu, Lilin, "Logserver monitor for managing log messages of applications" (2002). *Theses Digitization Project*. 2054.

https://scholarworks.lib.csusb.edu/etd-project/2054

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

LOGSERVER MONITOR FOR MANAGING

LOG MESSAGES OF APPLICATIONS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

٠.

Lilin Zhu

June 2002

LOGSERVER MONITOR FOR MANAGING

LOG MESSAGES OF APPLICATIONS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Lilin Zhu

June 2002

Approved by:

Dr. Richard Botting,	Chair,
Computer Science	·
Dr. Ernesto Gomez	

Dr. David Turner

May 15/2002

ABSTRACT

Logging represents an important component of a software development cycle. Almost every large software application includes its own logging or tracing API. This project creates a graphical user interface (GUI) to manage and display log information in a distributed environment.

This project uses Java "Swing" components to create the graphic user interface and uses Log4j APIs, a logging package for Java, to log messages from a distributed environment. The main user interface is composed of a menu bar and a tabbed window with three tabs: "File View," "Logging View," and "Email." The log messages can be displayed in the "Logging View" tab dynamically or can be saved in a log file.

The "File View" tab has a table and list area. The name and creation time of the existing log files can be displayed in the list area. The contents of the selected file can be displayed in the table. The table can be sorted by column and the user can turn on or off any column(s). The user can choose to display messages by types such as fatal, error, or info. The messages in the table can be color-coded by message type. The user can also change font

iii

and search the messages in the table. There are also printing, copy, and paste functions on the user interface.

The "Logging View" tab has a text area and list area. The list area lists the names of all clients that are currently connected or have been connected with the Log Server. The text area displays the log message received from all the clients. When the user selects a client from the list, the message from that client will be displayed in the new window.

The "Email" tab is designed for the user to send email to clients to inform them if errors occur.

iv

ACKNOWLEDGMENTS

I would like to thank Dr. Ernesto Gomez and Dr. David Turner for serving on my committee and their insightful comments on the draft of this report. I am particularly grateful to Dr. Richard Botting, my project advisor, for his patience, advice, guidance, detailed suggestions and corrections of the design, implementation, and report of this project.

I would also like to thank Mr. Jay Theodore of ESRI for getting me started on this project.

Last but not least, I would like to thank to my husband for his constant support for my continuing education in general and for this project in particular. He also spent many hours playing games with my daughter so I could work on this project.

v

TABLE OF CONTENTS

2

ABSTRACT	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER ONE: INTRODUCTION	
Purpose of the Project	1
Scope of the Project	2
Significance of the Project	3
Limitation of the Project	5
CHAPTER TWO: REVIEW OF RELATED WORK	
Related Logging Software	6
Silicon Graphics' System Log File Viewer	6
Another Graphic Log Viewer	7
Competing Logging Application Programming Interface	8
CHAPTER THREE: METHODOLOGY	
Software Development Environment	10
Development Environment	10
Running Platforms	11
Language Descriptions	11
Introduction	11
Java	11

vi

,

	Java	Logging Package	12
	Exte	nsible Markup Language	16
	Exte	nsible Style Sheet Language	
	Tran	sformation	17
Proj	ect S [.]	tructure	17
User	Inte	rface Description	18
	Main	Window	18
	The	"File View" Tab	21
		Open Log Files	21
		Sorting Messages by	24
			24
		Selecting a Message	26
		The "Edit" Menu	26
		The "Format" Menu	28
		The "Log Level" Menu	31
		The "Search" Menu	32
		The "Columns" Menu	36
	The	"Logging View" Tab	38
		Displaying Clients Information	38
		List of Clients	38
		Individual Client Window	39
	The	"Email" Tab	40
Clas	s Des	criptions	44
	Pack	ages Used in LogServer Monitor	44
	The LogS	New Classes Created for erver Monitor	45
	Proj User	Java Exter Tran Project S User Inte Main The The Class Des Pack The Logs	Java Logging Package Extensible Markup Language Extensible Style Sheet Language Transformation Project Structure User Interface Description Main Window The "File View" Tab Open Log Files Sorting Messages by Table Columns Selecting a Message The "Edit" Menu The "Edit" Menu The "Format" Menu The "Log Level" Menu The "Columns" Menu The "Columns" Menu The "Logging View" Tab Displaying Clients Information List of Clients Individual Client Window Packages Used in LogServer Monitor The New Classes Created for LogServer Monitor

CHAPTER FOUR: SUMMARY AND CONCLUDING REMARKS

..

Summary	49
Concluding Remarks	49
APPENDIX A: GLOSSARY OF TERMS	51
APPENDIX B: CLASS DESCRIPTIONS	53
REFERENCES	114

.

.

LIST OF TABLES

Table 1.	Log4j	Conversion	Character	Description	14
Table 2.	Color	Scheme			29

LIST OF FIGURES

Figure	1.	Sysmon Log Viewer Window	6
Figure	2.	Structure of LogServer Monitor	18
Figure	3.	File View Tab	19
Figure	4.	Logging View Tab	20
Figure	5.	Email Tab	20
Figure	6.	Open a File from the File Menu	21
Figure	7.	File Chooser Dialog (a)	22
Figure	8.	File Chooser Dialog (b)	22
Figure	9.	File List	23
Figure	10.	Select a File from the File List to Display	24
Figure	11.	Sort Table (a)	25
Figure	12.	Sort Table (b)	25
Figure	13.	Select a Message	26
Figure	14.	Сору	27
Figure	15.	Paste (a)	27
Figure	16.	Paste (b)	28
Figure	17.	Change Font (a)	28
Figure	18.	Change Font (b)	29
Figure	19.	Color Coding Messages	30
Figure	20.	Wrap Text	30
Figure	21.	Log Level Filter (a)	31
Figure	22.	Log Level Filter (b)	32
Figure	23.	Search Menu	32

Figure	24.	Find Message (a)	33
Figure	25.	Find Message (b)	34
Figure	26.	Find Message (c)	34
Figure	27.	Find Next Message (a)	35
Figure	28.	Find Next Message (b)	36
Figure	29.	Hide/Show Columns (a)	37
Figure	30.	Hide/Show Columns (b)	37
Figure	31.	Logging Client Messages	38
Figure	32.	A List of Clients	39
Figure	33.	Individual Client Window	39
Figure	34.	Send Email (a)	40
Figure	35.	Send Email (b)	41
Figure	36.	Send Email (c)	41
Figure	37.	Send Email (d)	42
Figure	38.	Send Email (e)	42
Figure	39.	Send Email (f)	43
Figure	40.	Send Email (g)	44

CHAPTER ONE

INTRODUCTION

Purpose of the Project

Logging is an important component of a software development cycle as well as for diagnostics of performance and monitoring of the software after deployment. It can provide precise context about a run of the application. This project is a graphical user interface for managing log information. The LogServer Monitor can help you easily view information about how an application is running and what kind of problem the application is having.

The idea of writing a Java LogServer Monitor started when the author collaborated with ESRI's ArcIMS Middleware team. ESRI (Environmental Systems Research Inc.) is a geographic information system (GIS) software company headquartered in Redlands, CA. ESRI wanted an application to display and manage the log messages created from its server programs in its Internet map server software, ArcIMS. This project started with that purpose in mind. Therefore, this is an attempt to solve a real-world software problem. ESRI's ArcIMS middleware is primarily developed in Java. Thus, Java is a logical choice for the application.

Scope of the Project

The LogServer Monitor provides a graphical user interface for the display and management of logged information from a distributed environment such as ESRI's ArcIMS Server components. The LogServer Monitor uses Java Swing components to create user interfaces and uses the Log4J APIs, a Java logging package to log the information. Log4j is an open source project, which allows developers to control which log statements are output with arbitrary granularity. It's fully configurable at runtime by using external configuration files. The logging behavior can be controlled by editing a configuration file without touching the application code [5].

In the application, the log messages can be either displayed dynamically in real time or written in the log files and then displayed in a table. This project has a LogServer Monitor window. From this window, a log request can be selected. The user can choose to view from the log file or view the log message dynamically.

The main user interface is composed of a menu bar and a tabbed window with three tabs: "File View," "Logging View," and "Email." The "File View" tab has a table and a list area. The name and creation time of the existing log files can be displayed in the list. There is an "Update

File" button on this tab. The user can update the file list by clicking on this button. The user can select which file to be displayed in the table from a list of available log files. The user can also open the log files from the file menu on the menu bar if he/she uses Microsoft Windows platform.

The "Logging View" tab has a list and a text area. The list displays the names of all clients that are currently or have been connected with log server. There is an "Update Clients" button on this tab. The user can update the client list by clicking on this button. All the log messages from all the clients are displayed in the text area dynamically. When the user selects a client from the list, that client's messages should be displayed in a new window dynamically.

The Email tab is designed for the user to send email through the network to inform the clients if errors occur.

Significance of the Project

Logging is important in software development because it enables developers to quickly see when a problem has occurred in the code without having to step through the code line-by-line. It is also crucial for the quality assurance personnel to create clear and specific

bug reports so developers can immediately zero in on the problem.

In addition, logging is crucial when an application is deployed. Once the application is deployed to a client site you can't start your debugger and you can't edit the code to determine what has gone wrong. Logging facilitates software servicing and maintenance at a client site by producing log reports for analysis by end users, system administrators, field service engineers, and software development teams.

However, a typical log file is a flat file with a list of hundreds of thousands or millions of lines of text. It is almost impossible for a user to wade through the text and find a particular piece of information. The user needs a set of tools to view what is stored in the log file. LogServer monitor is created precisely for this purpose. It provides a set of tools to allow the user to display log information in a variety of ways. So the user can easily use what is stored in the log files. In other words, the LogServer Monitor makes the inaccessible flat log file accessible and useful.

Limitation of the Project

There is a table in the "File View" Tab for displaying messages from log files. The table has a fixed number of columns for certain parts of a log message. This requires that the log files have certain format. The log file format should be defined as: %d %-5p [%t] %c - %m%n (refer to table 1. %: the beginning of the format. d: the date of the logging event. p: the priority of the logging event. t: the name of the thread that generates the logging event. c: the category of the logging event. m: logging event should be left justified to a width of five characters.). This is a limitation of this project.

Another limitation of this project is that the email function requires that the local host be running an SMPT server.

CHAPTER TWO

REVIEW OF RELATED WORK

Related Logging Software

Silicon Graphics' System Log File Viewer

Sysmon is a system log file viewer developed by Silicon Graphics Inc. (SGI) as a part of System Monitor, an error reporting system for the IRIX Interactive Desktop. The sysmon utility allows a user to browse the system log file generated by IRIX.

The sysmon utility simplifies the original eight system priorities into four priority levels: critical, error, warning, and info. The following diagram shows a sysmon System Log Viewer window:

n Carlos Contrator	Freq:	Date	Time	Hostname	Source	Message
)®	1,	Feb :14	00:45:03	iris .	unly:	Recoverable memory parity error detected by CPU at
Ø	1	Feb 14	00:45:03	iris	unix:	Recoverable memory parity error corrected by CPU a
6	4	Har 20	03:15:03	iris	unix:	Parity Error in SIMM S48¥
D.	1 °,	Mar 20	03:15:03	iris	unix:	Recoverable memory parity error detected by CPU at
D	1	Har 20	03:15:03	iris	unix:	Recoverable memory parity error corrected by CPU a
÷.,	2.	nay 9	21:00:40	iris.	unix:	HFS server silicon not responding still trying
	° 2	flay 9	22:58:34	iris 🐂 👻	unix:	NFS server silicon ok¥
D	1.	llov 26	15:27:00	iris	unix:	NARNING: clock gained 364 daysY
	3	Nov, 26	15:28:29	iris 📈 🖉	ypbind:	can't find address for aladdinY
12	2	Nov 26	16:04:54	iris	nediad:	Initialization of my address failed. Cannot conta
0	2	Nov 26	16:04:55	iris ·	xdn:	"Access file /war/X11/xdm/Xaccess, display iris unk
Ø	2	Nav 26	20:06:04	iris	INFO:	The system is shutting down.Y
ନ୍ତ	2	Nov 26	20:06:04	iris	INFO:	Please wait.Y
୭	2	Nov 26	20:06:08	iris	cyslogd:	going down on signal 15¥
ଲି	ʻ9 `	Nov 26	20:06:33	iris	unix:	2[WIRIX Release 5.3]
ම.	9	Nov 26	20:06:33	iris	unix:	Copyright 1987-1994 Silicon Graphics, Inc. Y
6	9	Nov: 26	20:06:33	iris	unix:	All Rights Reserved Y
	9	Nov 26	20:06:33	iris	Unix:	¥
	9	Nov 26	20:06:33	iris.	unis:	>>[92Perr_init: SusAD parity is enabled.Y
e de la	22	Sec.				
273	\$	5. 6. 92			S	·

Figure 1. Sysmon Log Viewer Window

The user can choose View, Filter and Sort options through the pull down menus on this window.

Some functions in the LogServer Monitor user interface were influenced by the sysmon utility. For example, The LogServer Monitor has a similar window as sysmon does.

Sysmon is a logging package for a specific operating system. By contrast, LogServer Monitor can be used to view and manage log information for any software application as long as it uses the Log file format. LogServer Monitor also uses XML technology but sysmon does not.

Another Graphic Log Viewer

Chainsaw is a graphic user interface for viewing log files. It also uses Log4J API. The log messages are dynamically displayed in a table. The interface allows a user to filter messages by their priorities. Chainsaw has influenced the functional design of the LogServer Monitor. For example, the LogServer Monitor borrowed the concept of filtering messages by their priorities from Chainsaw. However, Chainsaw has a very simple user interface with limited functions. LogServer Monitor has a more comprehensive user interface and many more functions for the user to explore the log information.

Competing Logging Application Programming Interface

Currently, there are two main competing Logging APIs for Java: Log4j, distributed under the Apache Software License, and Java Logging API by Sun Microsystems. The APIs of Log4j and Java Logging are similar in some respect. For example, they are both based on a named hierarchy. But there are also many differences. This project uses Log4j API because of the perceived advantages of Log4j over Java Logging API at the outset of this project. The following paragraphs list these advantages.

The Log4j is compatible with JDK1.1 and later. Java Logging API requires JDK 1.4.

In Log4j, it is very easy to change the priority of a category. By contrast, it is time consuming to change the priority of a category in Java Logging API.

In Log4j, appenders (output destination) and resource bundles are inherited from the hierarchy. In Java Logging API, a logger logs global handlers (appenders in Log4j). It does not inherit any handlers from the hierarchy.

Log4j has a set of priorities: FATAL, ERROR, WARN, INFO and DEBUG, which are clear and easy to understand. Java Logging API has a set of debugging levels: ALL,

SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST and off, which can be confusing.

Log4j has been around for several years and is being widely used in real world projects. In addition, Log4j has been ported to C++, Python and even C#. Therefore, Log4J is more mature and portable than Java Logging API.

CHAPTER THREE

METHODOLOGY

Software Development Environment

Development Environment

LogServer Monitor was developed using the following software:

- 1. Microsoft Windows operating system.
- 2. Sun Microsystems Java Development Kit(JDK) 1.3
- 3. Log4j (version 1.1.3)
- 4. Borland JBuilder4

Borland JBuilder4 is a GUI based Java development system. It is easy to compile and run Java programs in Jbuilder4. It has intuitive tools for creating new classes and setting up layouts. JBuilder4 can be used on Solaris, Linux, and Windows 98, NT, and 2000. JBuilder4 is hosted on JDK version 1.3 in order to take advantage of its debugging capabilities and enhanced client-side performance. However, one can still build applications using any previous versions of the JDK in Jbuilder4.

Jbuilder4 is one of the most comprehensive visual development environments for building applications,

applets, JSP/Servlets, JavaBeans, Enterprise JavaBeans and distributed J2EE applications for the Java 2 Platform.

Running Platforms

LogServer Monitor software can be run on the following platforms:

- 1. Windows operating system
- 2. Unix/Linux operating system

Language Descriptions

Introduction

The LogServer Monitor uses the Java API, especially the Java "Swing" API, to create the graphic user interface (GUI). It uses Log4j, a Java logging API, to send log messages from client programs to the LogServer Monitor. It uses XSLT style sheets to transform XML files to HTML on the fly when they are loaded in a Web Browser such as Internet Explorer 6.

<u>Java</u>

Java is one of the most popular software development languages. Java is a fully object-oriented language developed by Sun Microsystems. Java programs are capable of running on most popular computer platforms without the need for recompilation. Because of its portability,

multithreading, and networking capabilities, Java is being used for developing "middleware" to communicate between clients and databases and other server resources. Java designers eliminated manual memory allocation and deallocation. They introduced true arrays and eliminated pointer arithmetic. They eliminated multiple inheritances, replacing it with a new notion of interface. All these features in Java eliminated the possibility of creating code with the most common kind of bugs [2]. Sun Microsystems provides an implementation of Java 2 Platform called the Java 2 Software Development Kit(J2SDK), that includes the minimum set of tools you need to write software in Java [1].

Java Logging Package

Log4j, a popular logging package for Java, is an open source project. It allows the developer to control which log statements are output with arbitrary granularity. It is fully configurable at runtime using external configuration files [5]. Log4j has three main components:

- Categories: named entities
- Appenders: output destination
- Layouts: output format

These three types of components work together to log messages according to message type and priority, and to control how these messages are formatted and where they are reported. With Log4j, logging behavior can be controlled by editing a log configuration file, without touching the application binary [5]. Logged messages can be sent to different and multiple output destinations in a user-chosen format. The Log4j environment is configurable programmatically. It also supports configuration through files as well as XML documents [5].

Log4j makes it easy to name categories by software component. This can be accomplished by statically instantiating a category in each class, with the category name equal to the fully qualified name of the class [5].

In Log4j terminology, an appender means an output destination. Currently, appenders exist for the console, files, remote socket servers, etc. [5].

The layout is responsible for formatting the logging request according to the user's wishes. The PatternLayout, part of the Log4j distribution, lets the user specify the output format according to conversion patterns similar to the C programming language format strings. Each conversion

specifier starts with a percent sign (%) and is followed by optional format modifiers and a conversion character.

The conversion character specifies the type of data, e.g. category, priority, data, and thread name. The format modifiers control such things as field width, padding, left and right justification [5]. See table 1 for detail.

Conversion	Effect
Character	
8	Indicates that the next character
	or characters will specify that a
	conversion must take place before
	output.
С	Used to output the category of
	the logging event.
С	Used to output the fully
	qualified class name of the
	caller issuing the logging
	request.
d	Used to output the date of the
	logging event.

Table 1. Log4j Conversion Character Description

F	Used to output the file name						
	where the logging request was						
	issued.						
іт	Used to output location						
·	obed of subpat recation						
	information of the caller which						
	generated the logging event.						
L	Used to output the line number						
	from where the logging request						
	was issued.						
m	Used to output the application						
	supplied message associated with						
	the logging event.						
M	Used to output the Method name						
	where the logging request was						
	issued.						
n	Used to output the platform						
	dependent line separator						
	character or characters.						
p	Used to output the priority of						
	the logging event.						
r	Used to output the number of						
	Used to output the number of milliseconds elapsed since the						

.

•

,

	start of the application until
	the creation of the logging
	event.
·t	Used to output the name of the
	thread that generated the logging
	event.
x	Used to output the nested
	diagnostic context associated
	with the thread that generated
	the logging event.

The table is adapted from the Log4j Web site for Log4j technology:

http://jakarta.apache.org/log4j

Extensible Markup Language

XML, the Extensible Markup Language, is a meta-markup language for text documents. It has become popular because its structural markup allows documents to describe their own format and contents. It defines a generic syntax used to mark up data with simple, human readable tags. It provides a standard format for domains as diverse as web sites, electronic data interchange, vector graphics,

genealogy, real estate listings, object serialization, remote procedure calls, and voice mail systems [8]. Extensible Style Sheet Language Transformation

XSLT, Extensible Style sheet Language for Transformations, is a language which can transform XML files into a number of formats, HTML being one of them [4][8]. XSLT is based on finding parts of an XML document that match a series of predefined templates, and then applying transformation and formatting rules to each matched part.

Project Structure

The LogServer Monitor is composed of three parts: "File view," "Logging view," and "Email." The LogServer Monitor receives log information from clients through the socket and displays it either in "File View" or "Logging View." The "Email" provides a tool for the user to communicate with clients (Figure 2).



Figure 2. Structure of LogServer Monitor

User Interface Description

Main Window

The LogServer Monitor's main window consists of a menu bar and three tabs. The menu bar has six menus: "File", "Edit", "Format", "Log Level", "Search", and "Columns." The first tab is the "File View" tab. It has a list area, a button, a table, and a text area. It displays the log messages from the log files (Figure 3).

LogServerMonitor	e ² ***	**************************************	· · · · · · · · · · · · · · · · · · ·		• * * * *	
<u>File Edit Format</u>	Log Level Search	Colu <u>m</u> ns	i, i Stario			ter sense i
File view Logging Vi	ew Email	na in an	• .* 			
Update Log Files	Date	Time	Priority	Thread	Category	Messages

Figure 3. File View Tab

The second tab is the "Logging View" tab. It has a button, a list area, and a text area. It displays the log information dynamically (Figure 4).



Figure 4. Logging View Tab

The third tab is the "Email" tab. It has three text fields for the user to enter information about the sender ("From"), the receiver ("To"), and the message title ("Subject"). It has a text area for the user to type in the content of the email message (Figure 5).

LogServerMonitor					
File Edit Format Loy Level S	earch Colu <u>m</u> ns				
File view Logging View Email				W. W. M.	
rom:					
С.		www.kodowanaugu.comanaugu			
ubject:				100000000000000000000000000000000000000	
					,
			,		
				: -	
		and the second	<u></u>	7.45-3545	ar 686 - 77 - 7
	Si Si	end			

Figure 5. Email Tab

The "File View" Tab

<u>Open Log Files</u>. In the "File View" tab, the user can open log files from the "Open" menu item on the "File" menu. When the user clicks on the "Open" menu item, the open file dialog will pop up. The user can navigate to a directory to open log files. This function can be used only on the Windows platform (Figures 6, 7, and 8).

File Edit	Format Log	Level Searcl	h Columns	t the spectrum of the second	~ 14	t textost	Carl Strategy
Open 📐	origing View)	Email		1 N N		an a	
Print Exit	py Files \$	Date	Time	Priority	Thread	Category	Héssages
			,	·			

Figure 6. Open a File from the File Menu.

ile view Logging View	Email		승객은 이야기	4.1	v. 799 µ. 14 Manazarta	<u> </u>
Update Log Files	Date	Time	Priority T	'hread (Sategory	Message
Look in: Look in: Java Jbuilder4 aim95 All Users Application APPLOG BBSTORE	C WINDOWS C An C WINDOW MINDOW MINDOW	5			8883	
File <u>n</u> ame:]"	Open	
		CONTRACTOR OF THE OWNER OWNER OF THE OWNER OWNE	Carlow and a second second	unit state i su <u>e su se su se</u> se	4 M Thuến 1	£ 1

Figure 7. File Chooser Dialog (a)

le view Louging View	(Eugit				Sec. Sec.	and the second
Update Log Files	Date	Time	Priority	Thread	Category	Messages
						. 61
Open					et	
Tookin			<u> Kalindan</u> I	ন নি।	&	o
	1					
· 🗋 Genera	tor1.log					
🗋 Genera	tor2.log					
LogSer	ver.log			st. dé-	28. 6 19 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
🗋 LogSer	ver.log.1					
LogSen	ver.log.2					
Logsen	/er.xsl					
xmitest.	bit					
	balanning generation and an		uidi faiji uiti da a			
2 N 1	ىسىشىشىكى .	<u> </u>		len de la	<u> </u>	

Figure 8. File Chooser Dialog (b)

.

The user can choose from the list area which log file to be displayed. When the user clicks on the "Update Log Files" button on the "File View" tab, the names of all log files and the time the files were created will be displayed in the list area on the left hand side (Figure 9).

File Edit Format Log	Level Search Col	umns			
Update Log Files	Date Tin	ne Priority	Thread	Category	Messages
_ogServer.log_03/26/0 _ogServer.log.1_03/26					
Jenerator1.log 03/26/ Jenerator2.log 03/26/ LogServer.log.2 03/13					
	an a		en de la company		

Figure 9. File List

When the user clicks on a file in the list area, the content of the selected file will be displayed in the table on the right-hand side (Figure 10).

LogServerMonitor				1, 1,		. Eox
File Edit Format Lo	ig Level <u>S</u> eai	rch Colu <u>m</u> ns	And a function of		1	
File view Logging View	Email					
Update Log Files	Date 2002-03-26	Time 23:07:27,380	Priority INFO	Thread [main]	Category project.logs	Messages Listening o 🔺
LogServer.log 03/26/0 . LogServer.log.1 03/26	2002-03-26 2002-03-26	23:07:27,600 23:07:27,600	INFO DEBUG	[Thread-1] [Thread-1]	project.logs project.logs	Thread star Waiting for
Generator1.log 03/26/ Generator2.log 03/26/	2002-03-26 2002-03-26	23:10:28,250 23:10:28,300	DEBUG DEBUG	[Thread-1] [Thread-1]	project.logs project.logs	Socket[add
LogServer.log.2 03/13	2002-03-26	23:10:28,410 23:10:28,470		[Thread-1] [Thread-3]	project.logs	Waiting for Starting to
	2002-03-26	23:10:28,300	ERROR	(main) [main]	Generator1 Generator1	Unexpecte
	2002-03-26	23:10:28,470		[main] [main]	Generator1 Generator1	Hello, This
- 46571786.573	2002.02.26	22:10:20 620			Congrator1	
<u>(</u>)]					

Figure 10. Select a File from the File List to Display

Sorting Messages by Table Columns. The messages in the table can be sorted by table columns. This can be done by clicking the table column header. For example, if you click the column header "Priority", then the messages in the table will be sorted by the content in the priority column (Figures 11 and 12).
LogServerMonitor		direerriediked, taa	sur Billand an amhraidh an sfa	N 8	a 15	I	IX
<u>Filė Edit Format L</u>	og Level Sea	rch Colu <u>m</u> ns	\$ x * ;	1. 200		1999 - 1997 1997 -	1
File view Logging Vie	wEmail		1997 (M)				
Update Log Files LogServer.log 03/26/0 LogServer.log.1 03/26	Date 2002-03-26 2002-03-26 2002-03-26	Time 23:07:27,380 23:07:27,600 23:07:27,600	INFO INFO DEBUG	Thread [main] [Thread-1] [Thread-1]	Category project.logs project.logs project.logs	Messages Listening o Thread star Waiting for	4
Generator1.log 03/26/ Generator2.log 03/26/ LogServer.log.2 03/13	2002-03-26 2002-03-26 2002-03-26 2002-03-26 2002-03-26	23:10:28,260 23:10:28,300 23:10:28,410 23:10:28,470 23:10:28,300	DEBUG DEBUG DEBUG DEBUG	[Thread-1] [Thread-1] [Thread-1] [Thread-3] [main]	project.logs project.logs project.logs project.logs Generator1	Socket[add: Got a conn Waiting for Starting to This is Gen	
· · · · · · · · · · · · · · · · · · ·	2002-03-26 2002-03-26 2002-03-26 2002-03-26	23:10:28,410 23:10:28,410 23:10:28,470 23:10:28,470 23:10:28,470	ERROR FATAL INFO WARN	[main] [main] [main] [main]	Generator1 Generator1 Generator1 Generator1	Unexpecte Heré is a fa Hello, This Here is a W	
(1)		· .	-			<u></u>	

Figure 11. Sort Table (a)

🖉 LogServerMonito File Edit Format Log Level Search Columns File view, Logging View! Email Thread (main) Time Priority 23:11:48,820 DEBUG Message Hello there Category Generator2 Update Log Files Date 2002-03-26 2002-03-26 23:11:50,030 DEBUG 2002-03-26 23:11:50,090 DEBUG LogServer.log 03/26/0 (main) Generator2 Hello there, [main] Generator2 Hello there,. LogServer.log.1 03/26 23:11:50,090 DEBUG 23:11:51,960 DEBUG 23:11:53,010 DEBUG 23:11:53,330 DEBUG 23:11:53,440 DEBUG 23:38:05,790 DEBUG 10:05:24,370 DEBUG 10:05:24,370 DEBUG 2002-03-26 [main] Generator2 Hello there,. Generator1.log 03/26/ 2002-03-26 (main) Generator2 Hello there, Generator2.log 03/26/ 2002-03-26 (main) Generator2 Hello there,. LogServer.log.2 03/13 2002-03-26 (main) Generator2 Hello there, 2002-03-26 (Thread-1 project.logs... Waiting for 2002-03-27 Thread-1 project.logs. Waiting for 2002-03-26 23:10:28,410 ERROR (main) Generator1 Unexpecte. 2002-03-26 23:10:29,570 ERROR 2002-03-26 23:10:30,720 ERROR [main] Generator1 Unexpecte. (main) Generator1 Unexpecte 2002-03-26 23:10:31.870 ERROR [main] Generator1 Unexpecte **(** 36

Figure 12. Sort Table (b)

Selecting a Message. The user can select a message from the table by clicking on this message. Then the selected message will be highlighted. Meanwhile, the detailed message will be displayed in the text area below the table (Figure 13).

S LogServerMonitor		• •	1	s:	· · · · · .		X
<u>File Edit Format L</u>	oy Level <u>S</u> ear	rch Colu <u>m</u> ns					100
File view Longing View	v Email				e Çeşe		
Update Log Files	Date	s Time	Priority	- Thread	Category	Messages	2
	2002-03-26	23:07:27,380	INFO	[main]	project.logs	Listening o	
Fogseiver jog nalspin	2002-03-26	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star	
LogServer.log.1 03/26	2002-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for	
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logs	Socket[add	2
Generator2.log 03/26/	2002-03-26	23:10:28,300	DEBUG	[Thread-1]	project.logs	Got a conn	1 1 1
l occerver log 2 02/12	2002-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for	1
C0g081461.10g.2 03/13	2002-03-26	23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to	
	2002-03-26	23:10:28,300	DEBUG	[main]	Generator1	This is Gen	
	2002-03-26	23:10:28,410	ERROR	[main]	Generator1	Unexpecte	
	2002-03-26	23:10:28,410	FATAL	[main]	Generator1	Here is a fa	5
	2002-03-26	23:10:28,470	INFO	[main]	Generator1	Hello, This	્રે
	2002-03-26	23:10:28,470	WARN	[main]	Generator1	Here is a W	
	3002 02 28	22-10-20 520	WADN	. Impini	Gonorator1		
	Date: 2002-0	3-26					
	Time: 23:10:2	8,410					
	Priority: ERR()R					
l	Thread: [mair	1]					
	Category: Ge	nerator1					
1 I I I	Message: Ur	expected error	r!				

Figure 13. Select a Message

<u>The "Edit" Menu</u>. The "Edit" menu has "Copy" and "Paste" menu items. When the user clicks on the "Copy" menu item, the selected message from the table or the text area in the "Logging view" tab will be copied to the system clipboard. The copied message can be pasted in the text area on the "Email" tab or any other text editors such as Notepad by using their Paste function (Figures 14, 15, and 16).

🚭 LogServeiMonitor		* * V	,				I XI
File Edit Format L	og Level <u>S</u> ear	ch Colu <u>m</u> ns		S. Star		in the second	
File CONV D View	N) Email				\$_5.000	1997 - 1997 -	
Lit Paste	Date.	Time	Priority	Thread	Category	· Messages	\Box
/	2002-03-26	23:07:27,380	INFO	(main)	project.logs	Listening o	
LogServer.log 03/26/0	2002-03-26	23:07:27,800	INFO	[Thread-1]	project.logs	Thread star	
LogServer.log.1 03/26	2002-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for	
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logs	Socket[add	1 1
Generator2.log_03/26/	2002-03-26	23:10:28,300	DEBUG	[Thread-1]	project.logs	Got a conn	
LonServerion 2 03/12	2002-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for	
LUGUEIVEI.109.2 00/10	2002-03-26	23:10:28,470	DEBVG	[Thread-3]	project.logs	Starting to	11
	2002-03-26	23:10:28,300	DEBUG	(main)	Generator1	This is Gen	
	2002-03-26	23:10:28,410	ERROR	[main]	Generator1	Unexpecte	
	2002-03-26	23:10:28,410	FATAL	[main]	Generator1	Here is a fa	1.8
	2002-03-26	23:10:28,470	INFO	[main]	Generator1	Hello, This	- J
	2002-03-26	23:10:28,470	WARN	[main]	Generator1	Here is a W	
	ວັນບວ ບວ ວຍ	22.10.20 620		Imainl	Gonaratard		
	Date: 2002-03	3-26					
	Time: 23:10:2	8,410					
	Priority: ERRO	R					- 9
	Thread: [main	1					1
	Category: Ge	nerator1					
	Message: Un	expected error	1				

.

Figure 14. Copy

LogServerMonitor	in the same sector as a sector	
File Edit Format Log Level Search Columns		
File g View, Email		
rom Paste		
fo:		
Subject:		
		1
		l
		l
		1
Statistical Statistics	sug manamananananananananananananananananan	u Gini Çina alışı

.

Figure 15. Paste (a)

è

IS MEM	Logging View)	Email					398
im:							
: 			a tala haran da mu na haran ana ha		atabalaana soo kaasalaana soo		utmana: 177313
102-03-2	6 23.10.28 410 F	BROR [main]	Generator1	Unexpected erro	rð	tari takila karada sa sa karadi ti	ni:Hissorik
02-03-2	0 23.10.20,410 0	Trucon finality	Ceneratori		"1		

Figure 16. Paste (b)

The "Format" Menu. The "Format" menu has three menu items: "Font", "Color", and "Wrap."

When the user clicks on the "Font" menu item, a font dialog will pop up. The user can change font, size, and style for the text in the table (Figures 17 and 18).

LogServerMonilo				•	2 ⁰⁰⁰⁰ X,54 a **	na -yw, c	
File Edit Format	Log Lev	rel <u>S</u> ea	rch Colu <u>m</u> ns				
File view f Font	- 0	rail					
Lindate (Color	v í	Date	Time	Priority	Thread	Category	Messàges
127.84	+	-03-26	23:07:27,380	INFO	(main)	project.logs	Listening o 🔺
LogServer. [10 wrap	iext	-03-26	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star
LogServer.log.1 03/2	6 2002	-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for
Generator1.log 03/2	5/ 2 0 02	-03-26	23:10:28,250	DEBUG	(Thread-1)	project.logs	Socket[add
Generator2.log_03/2	2002	-03-26	23:10:28,300	DEBUG	[Thread-1]	project.logs	Got a conn
LooServer log 2 03/1	2002	-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for 🖪
Lugoerver.lug.z ug/1	2002	-03-28	23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to
	2002	-03-26	23:10:28,300	DEBUG	[main]	Generator1	This is Gen
	2002	-03-26	23:10:28,410	ERROR	[main]	Generator1	Unexpecte
	2002	-03-26	23:10:28,410	FATAL	(main)	Generator1	Here is a fa
	2002	-03-26	23:10:28,470	INFO	(main)	Generator1	Hello, This
I	2002	-03-26	23:10:28,470	WARN	[main]	Generator1	Here is a W
	2002	02.26	22:10:20 520	INADN.		. Ganaratari	L
	Date:	2002-0	3-26				
i i i i i i i i i i i i i i i i i i i	Time	: 23:10:	28,300				
	Priori	ty: DEBI	JG				
	Threa	ad: (mai	n]				
1	Cate	gory: Ge	enerator1				
▲ 1998/2011	Mess	age: Th	nis is Generato	r1.			

Figure 17. Change Font (a)

File view Logging Vi	ew Email	s, person		and a statement		
Update Log Files	1 Date	🕂 Time 🕔	Priority	Thread	Category	Messages
Enañésiésîna (10/00/0	2002-03-26	23:07:27,380	INFO	[main]	project.logs	Listening o
Logserver log Usizbio	2002-03-26	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star
LogServer.log.1 03/26	2002-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logs	Socket[add
Generator2 Iog_03/26/	2002-03-26	23:10:28,300	DEBUG	[Thread-1]	project.logs	Got a conn
Lageonarian 2 02/12	2002-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for
Logaerver.log.z oorra	2002-03-26	23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to
	2002-03-26	23:10:28,300	DEBUG	[main]	Generator1	This is Gen.,
	2002-03-26	23:10:28,410	ERROR	[main]	Generator1	Unexpecte
	2002-03-26	23:10:28,410	FATAL	[main]	Generator1	Here is a fa
Sec. dama	ac en conclud	22-10-20 170	UNISO.	Imain	Gonorator1,	Louo This
ان 🔁 Ch	ioose a Font	· · ·			ಾ ನಿನ್ನ ಕ್ಲೇಕ್ಸ್ ಪ್ರ ಕ್ಲಿಸ್ ಕ್ಲೇಕ್ಸ್ ಪ್ರ	a W.
Fon	t Abadi MT Con	densed Light	· · · ·	Style Plain	▼ Size 1	· ·
ABCD	EFGHUK	ideli destiti en dissenti		279 - 1 East of Station	<u></u>	
ahrde	fahiik					
1 124	rcznoo					
1234	567890					
Hello	World					

Figure 18. Change Font (b)

The user can also color code the text in the table by checking the "Color" menu item on the "Format" menu. The rows in the table will be colored according to the values in the "Priority" column. The color scheme is listed in the following table (Table 2, Figure 19).

Table 2. Color Scheme

	· · · · · · · · · · · · · · · · · · ·
Priority	Color
DEBUG	Black
INFO	Blue
WARN	Green
ERROR	Orange
FATAL	Red

🕞 LogServ	erMonitor	~	• • •	. *				· ; ; ; [80	\square
<u>File Edit</u>	Format	Log Le	vel <u>S</u> ear	rch Colu <u>m</u> ns					
File view	Font		nail .						
Update L	12 Color	-IN	Date	Time	Priority	Thread	Category	Messages *	\Box
	(T) Mran T	ovt.	-03-26	23:07:27,380	INFO	(main)	project.logs	Listening o	-
LogServer.i	ino an ab i	501	2-03-26	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star	
LogServer.l	og.1 03/26	200	2-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for	m
Generator1.	log 03/26/	200	2-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logs	Socket[add	
Generator2	.log 03/26/	200	2-03-26	23:10:28,300	DEBUG	(Thread-1)	project.logs	Got a conn	
LogSowerl	07 0 02/12	200	2-03-26	23:10:28,410	DEBUG	(Thread-1)	project.logs	Waiting for	
LUgoerver.	0y.2 03/13	200	2-03-26	23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to	
		200	2-03-26	23:10:28,300	DEBUG	[main]	Generator1	This is Gen	
		2:0	2-03-25	2110.28 \$10	ROAdd	(man)	Generator1	Une*pecte.	
		200	2-03-26	23:10:28,410	FATAL	[main]	Generator1	Here is a fa.,	
		200	2-03-26	23:10:28,470	INFO	[main]	Generator1	Hello, This	
		200	2-03-26	23 10 28,470	WARN	[main]	Generatori	Here is a W.,	
		2.3176	2.0.3.,28	22:10:28.6.0.	liaca 🖓 thi		Ganalari		.
		Date	2002-0	3-26					
ł		Time	e: 23:10:2	8,300					
ļ		Prio	rity: DEBL	JG					
1		Thre	ad: (mair	1]					1
1		Cate	gory: Ge	nerator1					
1	.)	Mes	sage: Th	is is Generato	r1				

Figure 19. Color Coding Messages

If the user checks the "Wrap" JCheckBoxMenuItem on the "Format" menu, the text in the table will be wrapped. This is for the purpose of displaying the whole message when the messages are too long to be fitted on a single row in the table's cells (Figure 20).

LogServerMonitor				⁻ ,		
File Edit Format L	og Level <u>S</u> ea	rch Colu <u>m</u> ns	N. A			
File view Font	nail				×.	
Update L 🗆 Color	Date	Time	Priority	Thread	Category	Messages
LogServer.I	ext -03-26	23:07:27,380	INFO	[main]	project.logser ver.LogServer	Listening on p ort 4445
LogServer.log.1 03/26 Generator1.log 03/26/	2002-03-26	23:07:27,600	INFO	(Thread-1)	project.logser ver.LoggingRe	Thread starte
Generator2.log 03/26/ LogServer.log.2 03/13	2002-03-26	23:07:27,600	DEBUG	(Thread-1)	project.logser ver.LoggingRe	Waiting for a d
	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logser ver.LoggingRe ceiver	Socket[addr= 127.0.0.1/127. 0.0.1,port=141 5,localport=44 45]
	2002-03-26 Date: 2002-0 Time: 23:10: Priority: FATA Thread: [mail Category: Ge	<u>23:10:28 300</u> 3-26 29,620 L n] enerator1	<u>DEBUG</u>	Thread-13	proiect logser	Got a connect
•	Message: H	ere is a fatal er	ror!			

Figure 20. Wrap Text

The "Log Level" Menu. The "Log Level" menu has "DEBUG", "INFO", "WARN", "ERROR", and "FATAL" JcheckBoxMenuItems, corresponding to the types of values in the "Priority" column in the table. The table shows only the type of messages that are checked on the Log Level menu. For example, when the "DEBUG" menu item is unchecked, this type of message will not be displayed. Initially, all these check boxes are checked. The user can uncheck or recheck the check boxes to display desired type(s) of messages (Figures 21 and 22).

🗟 LogServerMonitor	****	1., e - 4		\$ (\$ ⁽¹⁾)	÷	H e)×
File Edit Format	Log Level Sea	rch Colu <u>m</u> ns	S. S. S.				
File view Longing V	FATAL		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	1. 1. T. M.			
Lindate Log Files	ERROR ·	Time	Priority	Thread	Category	Messages	П
oprato 200	12 WARN	23:07:27,380	INFO	(main)	project.logs	Listening o	-
LogServer.log_03/26/	MINCO S	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star	
LogServer.log.1 03/2	Salaro 5	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for	P
Generator1.log 03/28	DEBUG	23:10:28,250	DEBUG	[Thread-1]	project.logs	Socket[add	1
Generator2 log_03/26/	2002-03-26	23:10:28,300	DEBUG	(Thread-1)	project.logs	Got a conn	11
LonConjor Lon 2, 03/12	2002-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for	1.
LODSelvelling'' novi a	2002-03-26	23:10:28,470	DEBUG	[Thread-3]	project.logs.,.	Starting to	11
	2002-03-26	23:10:28,300	DEBUG	(main)	Generator1	This is Gen	11
	2002-03-28	23 10 28,410	ERROR	[main]	Generator1	Unexpecte	1.1
	2002-03-26	23:10:28,410	FATAL	(main)	Generator1	Here is a fa	
	2002-03-26	23:10:28,470	INFO	[main]	Generator1	Hello, This	
	2002-03-26	23:10 28,470	WARN	[main]	Generator1	Hare is a W	_
	<u>00000.00.06</u>	122/10/20 420	hata on i	Incolai	Gaparatari		124
	Date: 2002-0	3-26					1
	Time: 23:10:2	28,300					
	Priority: DEBI	JG					
	Thread: [main	n]					
	Category: Gr	enerator1					
1	Message: Th	nis is Generator	r1.				

Figure 21. Log Level Filter (a)

🗑 LogServerMonitor					•****		ΙX
File Edit Format	Log Level S	earch Colu <u>m</u> ns	(alac Al		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	e est
File view Logging,V	🗹 FATAL		<u>, 11 - 11 - 11 - 11 - 11 - 11 - 11 - 11</u>				
Undate Log Files	ERROR	Time	Priority	Thread	Category	Messages	
	IZ WARN	5 23:07:27,380	INFO	[main]	project.logs	Listening o	
LogServer.log 03/26/	IN INFO	3 23:07:27,600	INFO	[Thread-1]	project.logs	Thread star	J
LogServer.log.1 03/2		2310-28,410	ERROR	main	(senarator)	Unexpecte	
Generator1.log 03/26	LA DEBUG	3 23:10:28,410	FATAL	(main)	Generator1	Here is a fa	
Generator2 Ing 113/26	2002-03-26	3 23:10:28,470	INFO	(main)	Generator1	Hello, This].]
LogCoworlog 2 02/1	2002-03-26	23:10:28,470	WARN	[main]	Generator1	Here is a W] []
_ogServer.log.2_03/1.3	2002-03-20	5 23.10:28,520	WARN	(main)	Generator1		
	2002-03-28	3 23:10:29,520	WARN	limaini	Generator I		
	2002-03-20	23:10:29,570	ERROR	[main]	Cenoratori	Unexpecte.	
	2002-03-28	3 23:10:29,620	FATAL	[main]	Generator1	Here is a fa	
	2002-03-26	23:10:29,620	INFO	[main]	Generator1	Hello, This	
	2002-03-26	3 23 10 29,630	WARN	[main]	Generator1	Here is a W.	
	2002 02.20	200000000000000000000000000000000000000	haramu .	itmans!	1 Canorstort	1	1000
	Date: 2002	-03-26			`		
	Time: 23:1	0:28,300	. ,				1
	Priority: DE	BUĢ		, '			
	Thread: (m	ain]					.
	Category:	Generator1	• .	- ,			1
•	Message:	This is Generato	r1.				

Figure 22. Log Level Filter (b)

The "Search" Menu. The "Search" menu has "Find" and "Find Next" menu items. From this menu, the user can search

the messages in the table by keyword (Figure 23).

LogServerMonitor							J×
File Edit Format Lu	og Level Sear	ch Columns	er er sie			\mathbb{E}_{n} is the \mathbb{E}_{n}	÷.,/
File view Longing May	Email - Fr	ul 💦 👘			1. S.		
Update Log Files	Dat Fi	nd Next	Priority	Thread	Category	Messages	Π
	2002-03-26	23:07:27,380	INFO	[main]	project.logs	Listening o	
LogServer.log_03/26/0	2002-03-28	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star	25
LogServer.log.1 03/26	2002-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for	H
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[[Thread-1]	project.logs	Socket[add	
Generator2 log_03(26(2002-03-26	23:10:28,300	DEBUG	[Thread-1]	project.logs	Got a conni	
LogPontoriog 2 02/12	2002-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for	
LUgaerver.iug.z 03/13	2002-03-26	23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to	
	2002-03-26	23:10:28,300	DEBUG	(main)	Generator1	This is Gen	
	2002-03-26	23:10:28,410	ERROR	[main]	Generator1	Unexpecte	
	2002-03-26	23:10:28,410	FATAL	[main]	Generator1	Here is a fa	
	2002-03-26	23:10:28,470	INFO	[main]	Generator1	Hello, This	
	2002-03-26	23:10:28,470	WARN	[main]	Generator1	Here is a W	
	2002.02.26	22:10:20 520	MADNI	limain1	Ganaratari		12
	Date: 2002-03	3-26 ·		· · ·			
	Time: 23:10:2	8,300				•	
	Priority: DEBU	IG ·				•	
	Thread: (main	1			•	•	
	Category: Ge	nerator1					
4	Message: Th	is is Generato	r1				ľ

Figure 23. Search Menu

When the user clicks the "Find" menu item, a pop-up window will let the user type in the keyword to start the search (Figure 24).

File Edit Format Lo	og Level – Sear	ch Colu <u>m</u> ns	. .		•		ΧI
File view Logging View	v Email		۰. ^۰ .			•	
Update Log Files	Date	Time .	Priority	Thread	Category	Messages	Ĩ
LogServer.log_03/26/0	2002-03-26	23:07:27,380	INFO	[main]	project.logs	Listening o	
LogServer.log.1 03/26	2002-03-26	23:07:27,800	DEBUG	Thread-11	project.logs.	Waiting for	
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logs.	. Socketladd	
Generator2.log 03/26/	2002-03-26	23:10:28,300	DEBUG	[Thread-1]	project.logs	. Got a conn	- international distance in the second se
LogServer.log.2 03/13	2002-03-	Input	concern and	$\Delta^* \omega$.	X ct.logs	. Waiting for	
	2002-03-:	Si Enter ti	ie text	s. Stevillarite	rator1	This is Gen	
6	2002-03-:	fatal	Sec. 20 8	<u>8., 1988</u> , N., NY,	rator1	Unexpecte	
	2002-03-:			inne <u>ie</u> g	rator1	Here is a fa	10000
	2002-03-3		OK Can	Cel 🕴	rator1	Hello, This	and a second
	2002-03-				rator1	neie is a w	-
	Date: 2002-03	3-26					
1000	Time: 23:10:2	8,300					
1000	Thread: Imain	1					on the second
	Catedory: Ge	nerator1					and the second se
4 100 500 100 100 100	Message: Th	is is Generato	r1,				Contraction of the

Figure 24. Find Message (a)

After the user types in the keyword and clicks the "OK" button, the message found will be highlighted in the table and the detailed message will be displayed in the text area below the table (Figures 25 and 26).

File view Logging View	w) Email			vý.				
Lindate Log Files	Date	Time	Priority	. Thread	C C	ategory	. Message	s h
abunco regiment	2002-03-26	23:07:27,380	INFO	[main]	pro	ject.logs	Listening	o 🔺
LogServer.log 03/26/0	2002-03-26	23:07:27,600	INFO	[Thread-1]	pro	ject.logs	Thread sta	ar 🖉
LogServer.log.1 03/26	2002-03-26	23:07:27,600	DEBUG	[Thread-1]	pro	ject.logs	Waiting for	r
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	pro	ject.logs	Socket[ad	d 🚺
Generator? Ion 03(26)	2002-03-26	23:10:28,300	DEBUG	Thread-11	pro	iectlogs	Got a conr	1 L
LogRowerlog 2 02/20/	2002-03-: 😹	Input			X	ctilogs	Waiting for	·)
Lugaener.lug.2 03/13	2002-03-:	e	1940 - 19 ¹ (ct.logs	Starting to	
	2002-03-:	S Enter th	ie text	and the second	2.7	rator1	This is Ge	nC
	2002-03-:	fatal			T	rator1	Unexpecte	6
	2002-03-	Indiana),	rator1	Here is a f	a
8	2002-03-1		ox Ca	too	de en	rator1	Hello, This	s]
. *	2002-03-:			ice)		rator1	Here is a \	N
8	2002 02 :		,'v		×	ratart	I	
. 8	Date: 2002-0	3-26						
	Time: 23:10:2	8,300						
	Priority: DEBL	JG						
	Thread: [mair	ı] [.]						
. 8	Category: Ge	nerator1	•					
	Maggage Th	ie ie Ganarsto	r1					

Figure 25. Find Message (b)

LogServerMonitor						·, soi
<u>File Edit Format L</u>	og Level Sea	rch Colu <u>m</u> ns	÷			
File view Logging Vie	w Email	1	1. AND 1.			
Undate Lon Files	Date	Time	Priority	Thread	Calegory	Messages
	2002-03-26	23:07:27,380	INFO	[main]	project.logs	Listening o
LogServer.log 03/26/0	2002-03-26	23:07:27,600	INFO	[Thread-1]	project.logs	Thread star
LogServer.log.1 03/26	2002-03-26	23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for
Generator1.log 03/26/	2002-03-26	23:10:28,250	DEBUG	[Thread-1]	project.logs	Socket[add
Generator? Iog. 03(26)	2002-03-26	23:10:28,300	DEBUG ·	[Thread-1]	project.logs	Got a conn
onConjoring 2 02/12	2002-03-26	23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for
Lugaerver.lug.z usits	2002-03-26	23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to
	2002-03-26	23:10:28,300	DEBUG	[main]	Generator1	This is Gen
	2002-03-26	23:10:28,410	ERROR	[main]	Generator1	Unexpecte
. 1	2002-03-26	23:10:28,410	FATAL	[main]	Generator1	Here is a fa
	2002-03-26	23:10:28,470	INFO	[main]	Generator1	Hello, This
. 4	2002-03-26	23:10:28,470	WARN	[main]	Generator1	Here is a W
	30.00 00 26	22:10:20.620	MAADN.	Imain	LConorator1	
	Date: 2002-0	3-26		•		
-	Time: 23:10:2	8,410	, .			(c) 1
	Priority: FATA	É .	•			
	Thread: Imair	าไ				1 A
2	Category Ge	nerator1		· · · ·		
T Residence Tr					1 e	

Figure 26. Find Message (c)

To find the next message containing the keyword, the user can click on the "Find Next" menu item. The next message found will be highlighted in the table and the detailed message will be displayed in the text area below the table (Figures 27 and 28).

SLogServerMonitor,			2000 - 2000 - 2000 2010 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000	****				
File Edit Format L	og Level	Search Columns	search Columns					
File view Logging View	w. Email	Fin <u>d</u> .			a anica.			
Update Log Files	Dat	Find Next	Priority	Thread	Category	Messages		
	2002-03	26 23:07:27,380	INFO	[main]	project.logs	Listening o 📥		
rodaeiverrind narsein!	2002-03	-26 23:07:27,600	INFO	[Thread-1]	project.logs	Thread star		
LogServer.log.1 03/26;	2002-03	26 23:07:27,600	DEBUG	[Thread-1]	project.logs	Waiting for		
Generator1.log 03/26/	2002-03	26 23:10:28,250	DEBUG	[Thread-1]	project.logs	Socketladd		
Generator2 Ing 03/26/	2002-03	-26 23:10:28,300	DEBUG	[Thread-1]	project.logs	Got a conn		
LogRomer log 2 02/12	2002-03	-26 23:10:28,410	DEBUG	[Thread-1]	project.logs	Waiting for		
Lugaerver.lug.z us/13	2002-03	26 23:10:28,470	DEBUG	[Thread-3]	project.logs	Starting to		
	2002-03	-26 23:10:28,300	DEBUG	[main]	Generator1	This is Gen		
l P	2002-03	-26 23:10:28,410	ERROR	[main]	Generator1	Unexpecte		
	2002-03	-26 23:10:28,410	FATAL	(main)	Generator1	Here is a fa		
l R	2002-03	26 23:10:28,470	INFO	[main]	Generator1	Hello, This		
	2002-03	26 23:10:28,470	WARN	[main]	Generator1	Here is a W		
	2002.02	26 22:10:20 620	MACAONI	Imain	Gonorotor1			
	Date: 20	02-03-26						
	Time: 23	8:10:28,410						
1	Priority:	FATAL						
1	Thread:	[main]						
	Category Generator1							
	Message	: Here is a fatal en	rori					

Figure 27. Find Next Message (a)

🖉 LogServerMonitor,	× 2	* e	· · · · · · · · ·		કાર્ટ નહેલ કોર્ગ	8 : <i>1</i>) E [IΧ
File Edit Format L	og Level Seal	ch Colu <u>m</u> ns	an in the second		1. 1. 1. 1. T. T.	a da ser de la compañía de	
File view Logging View	w Email		<u>.</u> National States		<u>) () () () () () () () () () () () () ()</u>		2 23
Update Log Files	Date 2002-03-26	Time 23:10:28,520	Priority	Thread (main)	Category Generatori	Messages	
LogServer.log 03/26/0	2002-03-26	23:10:28,520	WARN	[main]	Generator1		H
LogServer.log.1 03/26	2002-03-26	23:10:29,570	DEBUG	(main)	Generator1	This is Gen	
Generator1 Ion 03(26)	2002-03-26	23:10:29,570	ERROR	(main)	Generator1	Unexpecte	
Concreter2 leg 03/26/	2002-03-26	23:10:29,620	FATAL	[main]	Generator1	Here is a fa	1
Generatorz.log 03/20/	2002-03-26	23:10:29,620	INFO	[main]	Generator1	Hello, This	
LogServer.log.2 03/13	2002-03-26	23:10:29,620	WARN	(main)	Generator1	Here is a W	
	2002-03-26	23:10:29,680	WARN	(main)	Generator1		
	2002-03-26	23:10:29,680	WARN	(main)	Generator1		
	2002-03-26	23:10:30,670	DEBUG	[main]	Generator1	This is Gen	. 1
	2002-03-26	23:10:30,720	ERROR	[main]	Generator1	Unexpecte	11
	2002-03-26	23:10:30,720	FATAL	[main]	Generator1	Here is a fa	
	2002-03-26	23:10:30,780	INFO	[main]	Generator1	Hello, This	<u>-</u>
	Date: 2002-0	3-26					
	Time: 23:10:2	9.620					1
	Priority FATA						
	Thread: Imair	-					1
l li	Colonory Co	y norotori					
	acalegoly. Ge	neratori					1
	simessage: He	ere is a fatal er	ron]

Figure 28 Find Next Message (b)

<u>The "Columns" Menu</u>. The "Columns" menu has "Date", "Time", "Priority", "Thread", "Category" and "Messages" JcheckBoxMenuItems corresponding to the table column names. All these checkboxes are checked by default. When these checkboxes are unchecked, the column with name matching the checkbox name will be turned off. For example, if the "Date" menu item is unchecked, the first column in the table will not be displayed (Figures 29 and 30).

劉 LogServerMonitor			4	• =	
File Edit Format Lu	og Level <u>S</u> earch	Columns		1 A 1	
File view Logging View	v, Email	🗹 Date 🔊			
Update Log Files	Date	12 Time	ity Thread	Category	Messages .
LogServer.log*03/26/0	2002-03-26 23:0	Priority	[main] [Thread-1]	project.logs	Listening o A
LogServer.log.1 03/26	2002-03-26 23:0	M Inread	[Thread-1]	project.logs	Waiting for
Generator1.log 03/26/	2002-03-26 23:	Category	[Thread-1]	project.logs	Socket(add
Generator2.log 03/26/	2002-03-26 23:1	Messages	[Thread-1]	project.logs	Got a conn
LogServer.log.2 03/13	2002-03-26 23:1	0.28,410 DEBUG	[Inread-1]	project.logs	Starting for
	2002-03-26 23:1	0:28.300 DEBUG	Imain	Generator1	This is Gen
2	2002-03-26 23:1	0:28,410 ERROR	[main]	Generator1	Unexpecte
	2002-03-26 23:1	0:28,410 FATAL	[main]	Generator1	Here is a fa
	2002-03-26 23:1	0:28,470 INFO	[main]	Generator1	Hello, This
i i i i i i i i i i i i i i i i i i i	2002-03-26 23:1	10:28,470 WARN	[main]	Generator1	Here is a W
i i i i i i i i i i i i i i i i i i i	Date: 2002-03-26	10:20.520.WARN.		Gonoratori	
	Time: 23:10:29.67	'n			
	Priority; FATAL	-			
	Thread: [main]	-			
	Category: Genera	tor1			
(Message: Here is	a fatal error!	• • • •		

Figure 29. Hide/Show Columns (a)

LogServerMonitor				÷ • • •	s co x
File Edit Format L	og Level <u>S</u> earc	h Colu <u>m</u> ns	i serie in	na di i	
File view Logging Vie	w, Email		a second a second a second	1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 -	· · · ·
Update Log Files	Time	Priority	Thread *	Category	Messages ,
	23:07:27,380	INFO	(main)	project.logserv	Listening on p 🍝
LogServer.log_03/26/0	23:07:27,600	INFO	[Thread-1]	project.logserv	Thread started
LogServer.log.1 03/26	23:07:27,600	DEBUG	[Thread-1]	project.logserv	Walting for a c
Generator1.log_03/26/	23:10:28,250	DEBUG	[Thread-1]	project.logserv	Socket[addr=1
Generator2 log_03/26/	23:10:28,300	DEBUG	[Thread-1]	project.logserv	Got a connecti
LogCoverlog 2, 02/12	23:10:28,410	DEBUG	[Thread-1]	project.logserv	Waiting for a c
Lugoener.lug.z uaria	23:10:28,470	DEBUG	(Thread-3)	project.logserv	Starting to get
	23:10:28,300	DEBUG	[main]	Generator1	This is Gener
•	23:10:28,410	ERROR	[main]	Generator1	Unexpected er
	23:10:28,410	FATAL	[main]	Generator1	Here is a fatal
	23:10:28,470	INFO	[main]	Generator1	Hello, This is
	23:10:28,470	WARN	[main]	Generator1	Here is a War
	22:10:20 620	BAGAON1	Impint	Gonorator1	
•	Date: 2002-03-	26		•	
	Time: 23:10:29	,620			
	Priority: FATAL				
	Thread: [main]				
	Category: Gen	erator1			,
A	Message: Her	e is a fatal error!		•	

Figure 30. Hide/Show Columns (b)

The "Logging View" Tab

Displaying Clients Information. This tab is for dynamically displaying the log information. All of the messages logged from the clients are dynamically displayed in the text area as the messages are generated by a client program (Figure 31).

LogServerMonitor	1		i i i i i i i i i i i i i i i i i i i
<u>File Edit Format</u>	Log Level	Search Co	lumns
File view Logging \	liew Email	k	
Update Clients	03/26/02	11:10:32PM	WARN [main] Generator1 - Here is a Warnning message. 🔺
	03/26/02	11:10:32PM	WARN [main] Generator1 - null
	03/26/02	11:10:32PM	WARN [main] Generator1 - null
	03/26/02	11:10:33PM	DEBUG [main] Generator1 - This is Generator1.
	03/26/02	11:10:33PM	ERROR [main] Generator1 - Unexpected error!
	03/26/02	11:10:33PM	FATAL [main] Generator1 - Here is a fatal error!
	03/26/02	11:10:33PM	INFO [main] Generator1 - Hello, This is Generator1
	03/26/02	11:10:33PM	WARN [main] Generator1 - Here is a Warnning message.
	03/26/02	11:10:33PM	WARN [main] Generator1 - null
	03/26/02	11:10:33PM	WARN [main] Generator1 - null
	03/26/02	11:11:48PM	INFO [main] Generator2 - Hello from Generator2
	03/26/02	11:11:48PM	DEBUG [main] Generator2 - Hello there, this is Generator2.
	03/26/02	11:11:48PM	INFO (main) Generator2 - Hello, from Generator2
	03/26/02	11:11:48PM	ERROR [main] Generator2 - Here is a error message from
	03/26/02	11:11:48PM	FATAL [main] Generator2 - There is a fatal message!
	03/26/02	11:11:48PM	INFO [main] Generator2 - Hello!
	03/26/02	11:11:48PM	WARN [main] Generator2 - Here is a warnning message.
	03/26/02	11:11:48PM	WARN [main] Generator2 - Hello there 2
	4		

Figure 31. Logging Client Messages

List of Clients. All the names of the clients that are currently connected or have been connected with LogServer Monitor can be displayed in the list area on the left-hand side when the user clicks the "Update Clients" button (Figure 32).

LogServerMonitor			
File Edit Format :	Log Level	Search Co	lu <u>m</u> ns
File view Logging Vi	ew Email		
Update Clients N	03/26/02	11:10:32PM	· WARN [main], Generator1 - Here is a Warnning message.
Generator1	03/26/02	11:10:32PM	WARN [main] Generator1 - null
Generator2	03/26/02	11:10:33PM	DEBUG [main] Generator1 - This is Generator1.
	03/26/02	11:10:33PM	ERROR [main] Generator1 - Unexpected error
	03/26/02	11:10:33PM	FATAL [main] Generator1 - Here is a fatal error!
	03/26/02	11:10:33PM	WARN [main] Generator1 - Here is a Warnning message.
•	03/26/02	11:10:33PM	WARN [main] Generator1 - null
	03/26/02	11:10:33PM	WARN [main] Generator1 - null
18 T	03/26/02	11:11:48PM	DEBUG [main] Generator2 - Helio there, this is Generator2.
	03/26/02	11:11:48PM	INFO [main] Generator2 - Hello, from Generator2
	03/26/02	11:11:48PM	ERROR [main] Generator2 - Here is a error message from
· ·	03/26/02	11:11:48PM	INFO 1 main 1 Generator 2 - Hello!
1	03/26/02	11:11:48PM	WARN [main] Generator2 - Here is a warnning message.
	03/26/02	11:11:48PM	WARN [main] Generator2 - Hello there 2
J	ي ال	at and set in the	

Figure 32. A List of Clients

Individual Client Window. The user can open the

individual client window by clicking a client's name in the

list (Figure 33).

File Edit Format L	Level Search Columns	
File view Logging Vie	Email	
Update Clients	3/26/02 11:10:32PM WARN [main] Generator1 - Here is a Wamning message. 3/26/02 11:10:32PM WARN [main] Generator1 - null 3/26/02 11:10:32PM WARN [main] Generator1 - null 3/26/02 11:10:33PM DEBUG [main] Generator1 - This is Generator1. 3/26/02 11:10:33PM ERROR [main] Generator1 - Unexpected error1. 3/26/02 11:10:33PM FATAL [main] Generator1 - Here is a fatal error1. 3/26/02 11:10:33PM FATAL [main] Generator1 - Here is a Generator1. 3/26/02 11:10:33PM FATAL [main] Generator1 - Here is a Generator1. 3/26/02 11:10:33PM INFO [main] Generator1 - Here is a Warning message.	
	3/26 Generator1 3/26 03/26/02 11:10:28PM DEBUG [main] Generator1 - This is Generator1. 3/26 03/26/02 11:10:28PM ERROR [main] Generator1 - Unexpected error1. 3/26 03/26/02 11:10:28PM FATAL [main] Generator1 - Here is a fatal error1. 3/26 03/26/02 11:10:28PM FATAL [main] Generator1 - Here is a fatal error1. 3/26 03/26/02 11:10:28PM WARN [main] Generator1 - Here is a Warnning messag. 3/26 03/26/02 11:10:28PM WARN [main] Generator1 - null. 3/26 03/26/02 11:10:28PM DEBUG [main] Generator1 - This is Generator1. 03/26/02 11:10:28PM DEBUG [main] Generator1 - This is Generator1. 03/26/02 11:10:28PM ERROR [main] Generator1 - Unexpected error1. 03/26/02 11:10:29PM FATAL [main] Generator1 - Here is a fatal error1.	je.

Figure 33. Individual Client Window

The "Email" Tab

The user can send emails to clients through this tab. When the user clicks the "Send" button, a message window will pop up reporting whether the email has been successfully sent or not (Figures 34 and 35).

∰Log	ServerMo	nitor				and the second	
<u>F</u> ile	Edit For	nat <u>L</u> og Le	vel <u>S</u> earch	Columns		3.24 S. 1	
File vie	ew Logy	ing View 🗄	nail				
From:	Izhu@c	sci.csusb.ed	U				
To:	izhu@h	otmail.com					
Suhjec	t: Genera	tor1 has a er	ror		Luche and August and Au		
2002-1	03-26 23:1	0:28,410 ER	ROR [main] G	enerator1 Unexp	ected error!		
	oleoparijan nika		anna an	' Sent			

Figure 34. Send Email (a)

(종 LogS	erverMonitor
File Ec	dit Format Log Level Search Columns
File view	v. Longing View Email
From:	lzhu@csci.csusb.edu
To:	Izhu@hotmail.com
Subject:	Generator1 has a error
2002-03	3-26 23:10:28,410 ERROR [main] Generator1 Unexpected error!
	Send

Figure 35. Send Email (b)

When the email has been sent, the user clicks the "OK" button in the message dialog. The "Email" tab will be refreshed and all the text areas will be cleared (Figures 36° and 37).

File view	w Lagging View Email
From:	Izhu@csci.csusb.edu
To:	Izhu@hotmail.com
Subject:	Generator1 has a error
	Message Message has been sent.

Figure 36. Send Email (c)

LogServerMonitor			1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1		
Ede Edit Format Loy Level	Search Column	s			
File view Longing View Email	, <u>(</u> , , , , , , , , , , , , , , , , , , ,		Service and the		
From:					
Subject:					
and a subscription of the	and the second	Send		404 00100000 40001000	n Municipal de la company d La company de la company de

Figure 37. Send Email (d)

If the email has not been sent, a message window will pop up to report the error (Figure 38).

File viev	al Format Englevel Search Couloms v) Logging View Email
From:	Izhu@cscl.csusb.edu
lo:	Izhu@hotmail.com
sunject:	Generator1 has a error
	Mossage java.net.UnknownHostException:mailfust

Figure 38. Send Email (e)

After the user clicks on the "OK" button on the window reporting the error condition, the text in all the text areas on the "Email" tab will still be there so that it can be resent when the error condition is cleared (Figures 39 and 40).

Ele F	Suverion (co stant State and and a state of the second state of t
Filevie	NU Logging View] Email
From:	Izhu@csci.csusb.edu
To:	lzhu@hotmail.com
Subject	Generator1 has a error
	Message java.net.UnknownHostException: mailtost
	Send

Figure 39. Send Email (f)

لاق Log	verManilor	j ×
<u>File</u>	Format Log Level Search Columns	
File viev	Longing View Email	<u>.</u>
From	rhu@csci.csusb.edu	
То:	thu@hotmail.com	
Subject:	Senerator1 has a error	、
2002-03	26 23:10:28,410 ERROR [main] Generator1 Unexpected error!	
	· · · ·	
Se 3.	Send	

Figure 40. Send Email (g)

Class Descriptions

Packages Used in LogServer Monitor

As describe earlier, the LogServer Monitor uses the Java API, especially the Java "Swing" API, to create the graphic user interface (GUI). Specifically, the LogServer Monitor uses the following packages in the Java API:

- java.awt
- java.awt.datatransfer
- java.awt.event
- java.awt.geom
- java.awt.print
- java.net
- java.io

- java.util.StringTokenizer
- java.text.SampleDataFormat
- javax.swing
- javax.swing.table
- javax.swing.filechooser
- javax.swing.event
- java.util

The LogServer Monitor uses Log4J API, which includes:

- org.apache.log4j
- org.apache.log4j.Category
- org.apache.log4j.spi.LoggingEvent
- org.apache.log4j.Priority.

The New Classes Created for LogServer Monitor

The following classes have been newly created or adapted from other sources by the author for the LogServer Monitor:

- Data: stores the data contained in the table in a twodimensional array.
- Message: defines the log message object according to the components of the log messages. For example, a message object may contain date, time, priority, thread, category, message, etc.

- MessageVector: reads the messages from a selected log file, stores the messages to a vector, and writes the log file to an XML file.
- MyTableModel: defines the number of rows and columns in the table.
- Properties: Parses the information that is in the properties file.
- TableMap[6]: implements the TableModelListener interface in the javax.swing.table package.
- TableSorter[6]: provides sorting functions for the table. The sorting algorithm used is stable, which means that it does not switch rows when its comparison function returns 0 to denote that they are equivalent.
- ColumnHandler: enables the user to toggle on or off columns. It implements the ItemListener interface in the java.awt.event package.
- LogLevelHandler: enables the user to pick which types (priorities) of messages to be displayed. It implements ItemListener interface in the java.awt.event package.
- MyTableCellRenderer: renders the cells in the table. This class sets color of the text in the table. It is

a subclass of DefaultTableCellRenderer in the javax.swing.table.package.

- TextAreaRenderer: performs text wrapping in the table cell. This class implements the TableCellRenderer interface in the javax.swing.table package. It assigns a JTextArea control to each table cell and sets the JTextArea's Wrap property to true.
- EventLogging: creates a new thread when a new client is connected to accept log messages from clients.
- FontChooser[3]: inherits the JDialog class in the javax.swing package, displays a dialog and allows the user to select a font in any style and size from the list of available fonts on the system.
- ItemChooser[3]: presents the choices of "Font", "Style", and "Size" in a ComboBox.
- LoggingReceiver: listens for client connections.
- LogServerMonitor: the main class, manages all other classes and interfaces.
- ProcFrame: creates a new frame for displaying individual client's logging messages.

Please see the appendix B for a more complete. documentation of these classes such as their constructors

and methods. The sources for the adapted classes are also documented in Appendix B.

.

.

CHAPTER FOUR

SUMMARY AND CONCLUDING REMARKS

Summary

The LogServer Monitor is a system for displaying and monitoring log messages in a distributed system. It acts like a central server, which displays and monitors its clients' log information. The LogServer Monitor uses Java, Log4j, and XML technologies. The graphic user interface (GUI) of the LogServer monitor was designed to be user friendly. The log information from clients can be displayed dynamically in the LogServer Monitor GUI or saved on the server as log files which can be brought into the GUI. The user can send email to clients through the LogServer Monitor GUI to notify the client if there is a problem. The log files are also saved as XML files and an XML Style sheet (XSLT) file has been created for transforming the XML files to HTML on the fly.

Concluding Remarks

Logging is an important component of the software development and deployment. The logging captures the information such as configuration errors, performance bottlenecks, and bugs in the application. It can improve

the time of fixing a problem because the sooner an error detected, the cheaper it is to fix.

The LogServer Monitor provides a graphic user interface (GUI) to display and manage log information intuitively. The user can monitor how the application runs by examining the log information. The log information is also saved in log files to make it easy for the user to study at a later time.

These types of application are best written using Java programming language because it is easy to set up on almost any platform. Among many logging packages, Log4J is recommended because it is designed to be fast and flexible.

There are two limitations on the LogServer Monitor. First, the LogServer Monitor requires a fixed log file format. Otherwise, the log file cannot be displayed in the LogServer Monitor GUI properly. Secondly, the email function uses the mailto protocol. It requires that the local host be running an SMTP server. These limitations should be remedied in the future, improved, versions of this project.

APPENDIX A:

GLOSSARY OF TERMS

•

.

API Application programming interface.

GUI Graphic User Interface.

- IRIX An operating system developed by Silicon Graphic Ins.
- Java One of the most popular software development languages.

Log4j A logging package for Java.

SMTP Simple Mail Transport Protocol.

- XML Extensible Markup Language, a metamarkup language for text documents.
- XSLT Extensible Style sheet Language for Transformations, providing a standard way to transform XML files to a number of formats.

APPENDIX B:

.

,

CLASS DESCRIPIONS

...

,

.

project.logserver

Class Data

java.lang.Object

+--project.logserver.Data

public class Data

extends java.lang.Object

This class stores the data contained in the table in a two-

dimensional array.

Constructor Detail

Data

```
public Data()
```

Data default constructor.

Method Detail

getData

public java.lang.Object[][] getData(java.lang.String

fname, java.lang.String _dName)

Gets the data.

Parameters:

fname - the String representing the name of the input file to read from.

_dName - the String representing the

directory for the log files.

Returns:

an array of Object specifying the table Data.

getsize

```
public int getsize()
```

Gets the array size.

Returns:

an integer specifying the array size.

getMessage

public java.util.Vector getMessage()

Gets the Vector for the message.

Returns:

a Vector.

updateMessage

public java.lang.Object[][] updateMessage(java.util.HashSet

hs)

Gets the updated messages.

Parameters:

hs - the HashSet that contains the priority.

.

Returns:

an array of Object specifying the table Data.

••

project.logserver

Class EventLogging

java.lang.Object

+--project.logserver.EventLogging

public class EventLogging

extends java.lang.Object

implements java.lang.Runnable

This class processes a client connection and receives log information from clients.

Method Detail

setFrame

public void setFrame(javax.swing.JTextArea cp)

Sets a JTextArea for displaying all clients

information.

Parameters:

cp - the JTextArea.

run

public void run()

Listens for client connections.

Specified by:

run in interface java.lang.Runnable

project.logserver

Class FontChooser

java.lang.Object

```
+--java.awt.Component
```

```
+--java.awt.Container
```

```
+--java.awt.Window
```

+--java.awt.Dialog

```
|
+--javax.swing.JDialog
```

project.logserver.FontChooser

public class FontChooser

extends javax.swing.JDialog

This is a JDialog subclass that allows the user to select a font, in any style and size, from the list of available fonts on the system. The source code is from the "Java Examples in a Nutshell" by David Flanagan.

Constructor Detail

FontChooser

public FontChooser(java.awt.Frame owner)

FontChooser constructor.

Parameters:

owner - the Frame

Method Detail

getSelectedFont

public java.awt.Font getSelectedFont()

Gets the user's selection. If the user used the

"Cancel" button, this will return null.

Returns:

a Font.

getFontFamily

public java.lang.String getFontFamily()

Gets a font name.

Returns:

a String specifying the font name.

getFontStyle

public int getFontStyle()

Gets a font style.

Returns:

an integer specifying the font style.

getFontSize

public int getFontSize()

Gets a font size.

Returns:

an integer specifying the font size.

setFontFamily

public void setFontFamily(java.lang.String name)

Sets font.

Parameters:

name - the String representing a font name.

setFontStyle

public void setFontStyle(int style)

Sets font style.

Parameters:

style - the integer representing a font

style.

setFontSize

public void setFontSize(int size)

Sets font size.
Parameters:

size - the integer representing a font size.

setSelectedFont

public void setSelectedFont(java.awt.Font font)

Sets selected font.

Parameters:

font - the Font

changeFont

protected void changeFont()

Changes the Font

isModal

public boolean isModal()

Override this inherited method.

Overrides:

isModal in class java.awt.Dialog

Class ItemChooser

```
java.lang.Object
|
+--java.awt.Component
|
+--java.awt.Container
|
+--javax.swing.JComponent
|
+--javax.swing.JPanel
|
+--project.logserver.ItemChooser
```

public class ItemChooser

extends javax.swing.JPanel

This class presents the choices of "Font", "Style", and "Size" in ComboBox. This code is adapted from the "JAVA Examples in the Nutshell" by David Flanagan.

Constructor Detail

ItemChooser

public ItemChooser(java.lang.String name,

java.lang.String[] labels,

java.lang.Object[] values,

. . .

int defaultSelection)

ItemChooser constructor.

Parameters:

name - the String representing a choice name.

labels - the array of String for each choice option.

- values the array of Objects associated with each option.
- defaultSelect the integer that represents

a Font.

Method Detail

getName

public java.lang.String getName()

Gets choice name.

Overrides:

getName in class java.awt.Component

Returns:

a String representing the choice name.

getLabels

public java.lang.String[] getLabels()

Gets array of String for each choice option.

Returns:

an array of String.

getValues

public java.lang.Object[] getValues()

Gets array of an object associated with each option.

Returns:

an array of Object.

getSelectedIndex

public int getSelectedIndex()

Gets selected index.

Returns:

an integer.

getSelectedValue

public java.lang.Object getSelectedValue()

Gets selected value.

Returns:

an Object.

setSelectedIndex

public void setSelectedIndex(int selection)

Sets selected index.

Parameters:

selection - the integer representing the selected value.

select

protected void **select** (int selection)

Stores the new selected index and fires events to any registered listeners.

Parameters:

selection - the integer.

addItemChooserListener

```
public void addItemChooserListener(ItemChooser.Listener
```

listener)

Registers the event listener.

Parameters:

listener - the ItemChooser.Listener.

removeItemChooserListener

public void removeItemChooserListener(ItemChooser.Listener

listener)

Deregisters the event listener.

Parameters:

listener - the ItemChooser.Listener.

Class ItemChooser.Event

java.lang.Object

ł

```
+--java.util.EventObject
```

+--project.logserver.ItemChooser.Event

Enclosing class:

1

ItemChooser

public static class ItemChooser.Event

extends java.util.EventObject

This inner class defines the event type generated by

ItemChooser.

Constructor Detail

ItemChooser.Event

public ItemChooser.Event(ItemChooser source,

int selectedIndex,

java.lang.Object

selectedValue)

Event constructor.

Parameters:

source - the ItemChooser object.

66

selectedIndex - the integer for the index.

selectedValue - the object for selected

value.

Method Detail

getItemChooser

public ItemChooser getItemChooser()

Gets ItemChooser.

Returns:

an ItemChooser.

getSelectedIndex

public int getSelectedIndex()

Gets selected index.

Returns:

an integer.

getSelectedValue

public java.lang.Object getSelectedValue()

Gets selected value.

Returns:

an Object representing the selected value.

project.logserver

Interface ItemChooser.Listener

Enclosing class:

ItemChooser

public static interface ItemChooser.Listener

extends java.util.EventListener

This interface must be implementd by any object that wants to be notified when the current selection in an ItemChooser component changes.

•

Method Detail

.

itemChosen

public void itemChosen(ItemChooser.Event e)

Class LoggingReceiver

java.lang.Object

+--java.lang.Thread

+--project.logserver.LoggingReceiver

public class LoggingReceiver

extends java.lang.Thread

This class processes connections from the clients.

Method Detail

run

```
public void run()
```

Listens for client connections.

Overrides:

run in class java.lang.Thread

getHashtable

public java.util.Hashtable getHashtable()

Gets the hashtable that contains the individual clients window.

Returns:

a Hashtable.

getClientList

public java.util.Vector getClientList()

Gets the clients list.

Returns:

a Vector.

Class LogServerMonitor

```
java.lang.Object
|
+--java.awt.Component
|
+--java.awt.Container
|
+--java.awt.Window
|
+--java.awt.Frame
|
+--java.swing.JFrame
```

project.logserver.LogServerMonitor

public class LogServerMonitor

extends javax.swing.JFrame

implements java.awt.print.Printable

This is the main class for the LogServer Monitor.

Constructor Detail

LogServerMonitor

public LogServerMonitor()

LogServermonitor constructor.

Method Detail

search

public void **search**(int index,

java.lang.String input)

Searchs the message in the table.

Parameters:

index - the index of the row in the table on which to start the search.

getPosition

public int getPosition()

Gets the index of the row on which the message was found.

Returns:

an integer.

getFileList

public java.util.Vector getFileList()

Gets the log file list.

Returns:

a Vector.

createTablePane

protected javax.swing.JSplitPane createTablePane()

Creates a tab with a JTable, JList, JButton, and JTextArea.

createClientPane

protected javax.swing.JSplitPane createClientPane()

Creates a tab with a JTextArea, JButton, and JList.

createEmailPane

protected javax.swing.JPanel createEmailPane()

Creates the Email tab.

sendMail

public void sendMail()

Sends the email through the URL connection.

сору

```
public void copy()
```

Copys the selected text in the JTable or in the JTextArea to the system clipboard.

paste

```
public void paste()
```

Pastes the text from the system clipboard to the Email tab.

Print[6]

public int print(java.awt.Graphics g,

java.awt.print.PageFormat pageFormat,

int pageIndex)

throws java.awt.print.PrinterException

Print function.

Specified by:

print in interface java.awt.print.Printable

Parameters:

g - the Graphics. pageFormat - the PageFormat. pageIndex - the page index.

Returns:

an integer.

GetPageInfo [6]

public void getPageInfo(java.awt.Graphics g,

java.awt.print.PageFormat

pageFormat)

Gets page information.

Parameters:

g - the Graphics.

pageFormat - the PageFormat.

PrintTablePart [6]

public void printTablePart(java.awt.Graphics2D g2,

java.awt.print.PageFormat

pageFormat,

int rowIndex,

int columnIndex)

Prints the table.

Parameters:

g2 - the Graphic2D.
pageFormat - the PageFormat.
rowIndex - the table row index.
columnIndex - the table column index.

main

public static void main(java.lang.String[] args)

The main method.

Parameters:

args[0] - the property file.

Class LogServerMonitor.ColumnHandler

java.lang.Object

+--project.logserver.LogServerMonitor.ColumnHandler

Enclosing class:

LogServerMonitor

public class LogServerMonitor.ColumnHandler

extends java.lang.Object

implements java.awt.event.ItemListener

This class enables the user to toggle on or off columns.

Constructor Detail

LogServerMonitor.ColumnHandler

public LogServerMonitor.ColumnHandler()

Method Detail

itemStateChanged

public void itemStateChanged(java.awt.event.ItemEvent e)

This method is called if the user selects a

JCheckBoxMenuItem in the columnMenu.

Specified by:

itemStateChanged in interface

java.awt.event.ItemListener

.

Parameters:

e - the ItemEvent.

Class LogServerMonitor.LogLevelHandler

java.lang.Object

+--project.logserver.LogServerMonitor.LogLevelHandler

Enclosing class:

LogServerMonitor

public class LogServerMonitor.LogLevelHandler extends java.lang.Object implements java.awt.event.ItemListener This class enables the user to pick which types (priorities) of messages to be displayed.

Constructor Detail

LogServerMonitor.LogLevelHandler

public LogServerMonitor.LogLevelHandler()

Method Detail

itemStateChanged

public void **itemStateChanged**(java.awt.event.ItemEvent e)

This method is called if the user selects a

JCheckBoxMenuItem in the logMenu.

Specified by:

itemStateChanged in interface

java.awt.event.ItemListener

.

Parameters:

e - the ItemEvent.

,

Class LogServerMonitor.MyTableCellRenderer

```
java.lang.Object
 +--java.awt.Component
       +--java.awt.Container
              +--javax.swing.JComponent
                    +--javax.swing.JLabel
javax.swing.table.DefaultTableCellRenderer
project.logserver.LogServerMonitor.MyTableCellRenderer
Enclosing class:
```

LogServerMonitor

public class LogServerMonitor.MyTableCellRenderer
extends javax.swing.table.DefaultTableCellRenderer

This class renders the cells in the table and sets the color of the text in the table.

Constructor Detail

LogServerMonitor.MyTableCellRenderer

public LogServerMonitor.MyTableCellRenderer()

Method Detail

getTableCellRendererComponent

public java.awt.Component

getTableCellRendererComponent(javax.swing.JTable

table, java.lang.Object value, boolean isSelected, boolean hasFocus, int row, int column)

Returns the table cell renderer.

Overrides:

getTableCellRendererComponent in class

javax.swing.table.DefaultTableCellRenderer

Parameters:

table - the JTable. value - the value to assign to the cell at [row, column]. isSelected - true if cell is selected. hasFocus - true if cell has focus. row - the row of the cell to render.

81

column - the column of the cell to render.

•

.

·

Class LogServerMonitor.OrigCellRenderer

```
java.lang.Object
 +--java.awt.Component
        +--java.awt.Container
              +--javax.swing.JComponent
                    +--javax.swing.JLabel
javax.swing.table.DefaultTableCellRenderer
project.logserver.LogServerMonitor.OrigCellRenderer
Enclosing class:
     LogServerMonitor
public class LogServerMonitor.OrigCellRenderer
extends javax.swing.table.DefaultTableCellRenderer
```

This class renders the cells in the table.

Constructor Detail

LogServerMonitor.OrigCellRenderer

public LogServerMonitor.OrigCellRenderer()

Method Detail

getTableCellRendererComponent

public java.awt.Component

getTableCellRendererComponent(javax.swing.JTable table,

java.lang.Object value, boolean isSelected,

boolean hasFocus, int row, int column)

Returns the table cell renderer.

Overrides:

getTableCellRendererComponent in class

javax.swing.table.DefaultTableCellRenderer

Parameters:

table - the JTable. value - the value to assign to the cell at [row, column]. isSelected - true if cell is selected. hasFocus - true if cell has focus. row - the row of the cell to render. column - the column of the cell to render.

84

Class LogServerMonitor.TextAreaRenderer

```
java.lang.Object
|
+--java.awt.Component
|
+--java.awt.Container
|
+--javax.swing.JComponent
|
+--javax.swing.text.JTextComponent
|
+--javax.swing.text.JTextArea
|
+--javax.swing.JTextArea
```

project.logserver.LogServerMonitor.TextAreaRenderer

Enclosing class:

LogServerMonitor

public class LogServerMonitor.TextAreaRenderer

extends javax.swing.JTextArea

implements javax.swing.table.TableCellRenderer

This calss performs text wrapping in the table cell.

Constructor Detail

LogServerMonitor.TextAreaRenderer

public LogServerMonitor.TextAreaRenderer()

Method Detail

getTableCellRendererComponent

public java.awt.Component

getTableCellRendererComponent(javax.swing.JTable table,

java.lang.Object value, boolean isSelected, boolean

hasFocus, int row, int column)

Returns the table cell renderer.

Specified by:

getTableCellRendererComponent in interface
javax.swing.table.TableCellRenderer

Parameters:

table - the JTable. value - the value to assign to the cell at [row, column]. isSelected - true if cell is selected. hasFocus - true if cell has focus. row - the row of the cell to render. column - the column of the cell to render.

Class Message

java.lang.Object

+--project.logserver.Message

public class Message

extends java.lang.Object

This class defines the log message object according to the components of the log messages. For example, a message object may contain date, time, priority, thread, category, message, etc.

Constructor Detail

Message

```
public Message()
```

Message default constructor.

Message

public Message(java.lang.String _date,

java.lang.String _time, java.lang.String _priority, java.lang.String _tName, java.lang.String _cName,

87

```
java.lang.String m)
```

Message constructor.

Parameters:

category.

m - the String representing the message.

Method Detail

setDate

public void setDate(java.lang.String _date)

Sets the date.

Parameters:

_date - the String representing the date.

setTime

public void setTime(java.lang.String _time)

Sets the time.

Parameters:

_time - the String representing the time.

setPriority

public void setPriority(java.lang.String _priority)

Sets the priority.

Parameters:

_priority - the String representing the

priority.

setThread

public void setThread(java.lang.String _tName)

Sets the thread.

Parameters:

_tName - the String representing the thread.

setCategory

public void setCategory(java.lang.String _cName)

Sets the category.

Parameters:

_cName - the String representing the

category.

setMessage

public void setMessage(java.lang.String m)

Sets the message.

Parameters:

m - the String representing the message.

getDate

public java.lang.String getDate()

Gets the date.

Returns:

a String representing the date.

getTime

public java.lang.String getTime()

Gets the time.

Returns:

a String representing the time.

getPriority

public java.lang.String getPriority()

Gets the priority.

Returns:

a String representing the priority.

getThread

public java.lang.String getThread()

Gets the thread.

Returns:

a String representing the thread.

getCategory

public java.lang.String getCategory()

Gets the category.

Returns:

a String representing the category.

getMessage

public java.lang.String getMessage()

Gets the message.

Returns:

a String representing the message.

· .

τ.

Class MessageVector

java.lang.Object

+--project.logserver.MessageVector

public class MessageVector

extends java.lang.Object

This class reads the messages from a selected log file, stores the messages to a vector, and writes the log file to an XML file.

Constructor Detail

MessageVector

public MessageVector()

MessageVector default constructor.

Method Detail

readFile

public void readFile(java.lang.String file_Name,

java.lang.String dir_Name)

Reads log information from a log file, puts the information in a Vector, and writes the log information to an XML file.

Parameters:

getVector

public java.util.Vector getVector()

Returns a Vector that stores log messages.

Returns:

a Vector.

Class MyTableModel

java.lang.Object

```
' +--javax.swing.table.AbstractTableModel
```

+--project.logserver.MyTableModel

public class MyTableModel

extends javax.swing.table.AbstractTableModel

This class defines the table model.

Constructor Detail

MyTableModel

```
public MyTableModel()
```

The MyTablemodel default construtor.

Method Detail

setTable

public void setTable(java.lang.String fileName,

java.lang.String dName)

Sets the table data.

Parameters:

fileName - the String representing the name

of the input file to read from.

dName - the String representing the

directory for the log files.

resetTable

public void resetTable(java.util.HashSet hs)

Resets the table data.

Parameters:

hs - the HashSet.

getColumnCount

public int getColumnCount()

Gets the number of columns in the model.

Overrides:

getColumnCount in class

javax.swing.table.AbstractTableModel

Returns:

an integer.

getRowCount

public int getRowCount()

Gets the number of rows in the model.

Overrides:

getRowCount in class

javax.swing.table.AbstractTableModel

Returns:

an integer.

getColumnName

public java.lang.String getColumnName(int col)

Gets the name of the column at col.

Overrides:

getColumnName in class

javax.swing.table.AbstractTableModel

Parameters:

col - the index of the column.

Returns:

a String representing the column name.

getValueAt

public java.lang.Object getValueAt(int row,

int col)

Gets the value for the cell at col and row.

Overrides:

getValueAt in class

javax.swing.table.AbstractTableModel

Parameters:
row - the row whose value is to be queried.

col - the column whose value is to be

queried.

Returns:

a value Object at the specified cell.

· .

getColumnClass

public java.lang.Class getColumnClass(int c)

Gets the most specific superclass for all the cell values in the column.

Overrides:

getColumnClass in class

javax.swing.table.AbstractTableModel

Parameters:

c - the index of the column.

Returns:

a common ancestorClass of the object values in the model.

isCellEditable

public boolean isCellEditable(int row,

int col)

Returns true if the cell is editable, false otherwise.

Overrides:

isCellEditable in class

javax.swing.table.AbstractTableModel

Parameters:

row - the row whose value to be queried.

col - the column whose value to be queried.

Returns:

true if the cell is editable.

setValueAt

public void setValueAt(java.lang.Object value,

int row,

```
int col)
```

Sets the value in the cell at col and row to value.

Overrides:

setValueAt in class

javax.swing.table.AbstractTableModel

Parameters:

value - the new value.

row - the row whose value is to be changed.

col - the column whose value is to be

changed.

```
project.logserver
```

Class ProcFrame

```
java.lang.Object
|
+--java.awt.Component
|
+--java.awt.Container
|
+--java.awt.Window
|
+--java.awt.Frame
|
+--java.swing.JFrame
|
+--javax.swing.JFrame
```

project.logserver.ProcFrame

public class ProcFrame

extends javax.swing.JFrame

This class creates a new frame for displaying individual client's logging messages.

project.logserver

Class Properties

java.lang.Object

+--project.logserver.Properties

public class Properties

extends java.lang.Object

This class parses the information that is in the property file.

Constructor Detail

Properties

```
public Properties()
```

Method Detail

parseProperties

public void parseProperties(java.lang.String filename)

Parses the information that is in the property file.

Parameters:

filename - the String representing the name of the input file to read from.

project.logserver

Class TableMap

```
java.lang.Object
|
+--javax.swing.table.AbstractTableModel
|
+--project.logserver.TableMap
```

Direct Known Subclasses:

TableSorter

public class TableMap

extends javax.swing.table.AbstractTableModel

implements javax.swing.event.TableModelListener

This class implements TableModel by routing all requests to its model, and TableModelListener by routing all events to its listeners. The source code is from the "The Java Tutorial" by Philip Milne.

Constructor Detail

TableMap

```
public TableMap()
```

The TableMap default constructor.

Method Detail

getModel

public javax.swing.table.TableModel getModel()

Gets the TableModel.

Returns:

a TableModel.

setModel

public void setModel(javax.swing.table.TableModel model)

Sets the TableModel.

Parameters:

model - the TableModel.

getValueAt

public java.lang.Object getValueAt(int row,

int col)

Gets the value of a specific table cell.

Overrides:

getValueAt in class

javax.swing.table.AbstractTableModel

Parameters:

row - the row whose value is to be queried.

col - the column whose value is to be

queried.

Returns:

an Object at the specified cell.

setValueAt

public void setValueAt(java.lang.Object value,

int row,

int column)

Sets the value in the cell at col and row to value.

Overrides:

setValueAt in class

javax.swing.table.AbstractTableModel

Parameters:

value - the new value. row - the row whose value is to be changed. col - the column whose value is to be changed.

getRowCount

public int getRowCount()

Gets the number of rows in the model.

Overrides:

getRowCount in class

javax.swing.table.AbstractTableModel

Returns:

an integer represents the number of rows in the model.

getColumnCount

public int getColumnCount()

Gets the number of columns in the model.

Overrides:

getColumnCount in class

javax.swing.table.AbstractTableModel

Returns:

an integer represents the number of columns in the model.

getColumnName

public java.lang.String getColumnName(int aColumn)

Gets the name of the column at aColumn.

Overrides:

getColumnName in class

javax.swing.table.AbstractTableModel

Parameters:

aColumn - the index of the column.

Returns:

a String representing the column name.

getColumnClass

public java.lang.Class getColumnClass(int aColumn)

Gets the most specific superclass for all the cell values in the column.

Overrides:

getColumnClass in class

javax.swing.table.AbstractTableModel

Parameters:

aColumn - the index of the column.

Returns:

a class.

isCellEditable

public boolean isCellEditable(int row,

int column) ·

Returns true if the table editable, otherwise false.

Overrides:

isCellEditable in class

javax.swing.table.AbstractTableModel

Parameters:

row - the row whose value to be queried. column - the column whose value to be

queried.

Returns:

true if the cell is editable.

tableChanged

public void tableChanged(javax.swing.event.TableModelEvent

e)

Forwards all events to all the listeners by default.

Specified by:

tableChanged in interface

javax.swing.event.TableModelListener

Parameters:

e - TableEvent.

```
project.logserver
```

Class TableSorter

```
java.lang.Object
|
+--javax.swing.table.AbstractTableModel
|
+--project.logserver.TableMap
|
+--project.logserver.TableSorter
```

public class TableSorter

extends TableMap

```
A sorter for TableModels. The source code is from the "The Java Tutorial" by Philip Milne.
```

Constructor Detail

TableSorter

```
public TableSorter()
```

TableSort default constructor.

TableSorter

public TableSorter(javax.swing.table.TableModel model)

TableSorter constructor.

Parameters:

model - the Tablemodel.

Method Detail

setModel

public void setModel(javax.swing.table.TableModel model)

۰.

Sets the tablemoel.

Overrides:

setModel in class TableMap

Parameters:

model - the TableModel.

compareRowsByColumn

public int compareRowsByColumn(int row1,

int row2,

int column)

Compares the table rows by the current column. The returned integer of -1, means that the value in row1 is less than that in row2; 1 means that the value in row1 is greater than that in row2; and 0 means that the value in row1 is equal to that in row2.

Parameters:

row1 - the first row to be compare with. row2 - the second row to be compare with. column - the indext of the current column.

Returns:

an integer.

compare

public int compare(int row1,

int row2)

Compares values in two rows in the table. Returned value of 0 represents ascending order.

Parameters:

row1 - the first row to be compare with.

row2 - the second row to be compare with.

Returns:

Ο.

reallocateIndexes

public void reallocateIndexes()

Sets up a new array of indexes for the number of elements for the new data model.

tableChanged

public void tableChanged(javax.swing.event.TableModelEvent

e)

Notifies that the table model changed.

Overrides:

tableChanged in class TableMap

Parameters:

e - the TableModelEvent.

checkModel

.

public void checkModel()

Checks if the table model has been changed.

sort

public void sort(java.lang.Object sender)

Sorts the table.

Parameters:

sender - the Object.

shuttlesort

public void shuttlesort(int[] from,

int[] to,

int low,

int high)

Sorts the data in the table.

Parameters:

from - the first half of array.
to - the end last half of array.
low - the beginning index.
high - the end index.

swap

getValueAt

public java.lang.Object getValueAt(int row,

int col)

Gets the value for the cell at col and row.

Overrides:

getValueAt in class TableMap

Parameters:

row - the row whose value is to be queried.

col - the column whose value is to be

queried.

Returns:

a value Object at the specified cell.

setValueAt

public void **setValueAt**(java.lang.Object value,

int row,

int col)

Sets the value in the cell at col and row to value.

Overrides:

setValueAt in class TableMap

Parameters:

value - the new value. row - the row whose value is to be changed. col - the column whose value is to be changed.

sortByColumn

public void sortByColumn(int column)

Sort the table by column.

Parameters:

column - the table column index.

sortByColumn

public void sortByColumn (int column,

boolean ascending)

Sorts the table by column.

Parameters:

column - the table column index.

ascending - true if by the asccending order.

addMouseListenerToHeaderInTable

public void

addMouseListenerToHeaderInTable(javax.swing.JTable table)

1

Add a mouse listener to the Table to trigger a table sort when a column heading is clicked in the JTable.

Parameters:

table - the JTable.

.

.

REFERENCES

- [1] Deitel \$ Deitel, "Java How to Program", Prentice Hall, Third Edition, 1999.
- [2] Cay S. Hostmann & Gary Cornell "Core Java", The Sun Microsystems Press, 2000.
- [3] David Flanagan, "Java Example in a Nutshell", O'Reilly, 2000
- [4] "XML Programming", Hands On Technology Transfer, Inc., 2000.
- [5] Log4j Web site for Log4j technology http://jakarta.apache.org/log4j
- [6] The Source for Java Technology Web Site http://java.sun.com/docs/books/tutorial/uiswing
- [7] The chainsaw web site

http://www.puppycrawl.com/chainsaw.

[8] David Hunter, "Beginning XML", Wrox Press Ltd, 2000